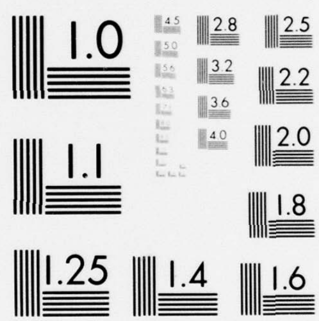
AD-A064 140 SCHOOL OF AEROSPACE MEDICINE BROOKS AFB TEX F/G 9/2
A COMPUTERIZED OPTICAL SCAN DATA FORM GENERATOR.(U)
DEC 78 J W SLAUGHTER, R L MEDINA
UNCLASSIFIED SAM-TR-78-34 NL

| OF |

AD
A064140

END
DATE
FILMED
4-79
DDC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

Report SAM-TR-78-34

A COMPUTERIZED OPTICAL SCAN DATA FORM GENERATOR

John W. Slaughter, B.A.
Richard L. Medina, B.S.

December 1978

Final Report for Period July - December 1978

Approved for public release; distribution unlimited.

USAF SCHOOL OF AEROSPACE MEDICINE
Aerospace Medical Division (AFSC)
Brooks Air Force Base, Texas 78235

NOTICES

This final report was submitted by personnel of the Repositories Branch, Data Sciences Division, USAF School of Aerospace Medicine, Aerospace Medical Division, AFSC, Brooks Air Force Base, Texas, under job order 9993XXXX.

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

JOHN W. SLAUGHTER, B.A.
Project Scientist

WILLIAM E. NIXON, B.S.
Supervisor

LAWRENCE J. ENDERS
Colonel, USAF, MC
Commander

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: SAM-TR-78-34	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): A COMPUTERIZED OPTICAL SCAN DATA FORM GENERATOR	5. TYPE OF REPORT & PERIOD COVERED: Final Rept. July - December 1978
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): John W. Slaughter, B.A.; Richard L. Medina, B.S.	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: USAF School of Aerospace Medicine (BRP), Aerospace Medical Division (AFSC), Brooks Air Force Base, Texas 78235	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62202F, 9993-XX-XX
11. CONTROLLING OFFICE NAME AND ADDRESS: USAF School of Aerospace Medicine (BRP), Aerospace Medical Division (AFSC), Brooks Air Force Base, Texas 78235	12. REPORT DATE: December 1978
	13. NUMBER OF PAGES: 21
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 23p.	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

ACCESSION for	
NTIS	White Section ☒
DDC	Buff Section ☐
UNANNOUNCED	☐
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

OPSCAN
CALCOMP Basic Software
Cartesian coordinate plane
Ordered pairs
Grid origin

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A system for generating OPSCAN readable data collection forms is described which enables free form design of computer input documents. The main component of the system is a Fortran IV program utilizing two subroutines from the CALCOMP Basic Software. A sample application with illustrations is provided.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 01 29 050 317000

A COMPUTERIZED OPTICAL SCAN DATA FORM GENERATOR

INTRODUCTION

In some data collection efforts it is beneficial to utilize an optical scanning machine because of its speed and reliability. However, due to the time-consuming programming involved in the form design, an optical reader, historically, has been useful only for production applications with a fairly static input design. If a user was able to meet the requirement of having a fairly well defined data collection application such as with a repository, the use of optical scanning equipment proved to be an asset to the overall system design.

Generally, when optical scanning equipment was used, a data monitor marked a prepared form which was usually designed by the OPSCAN Corporation and readable by OPSCAN equipment. This is quite common and is frequently used at USAFSAM. Drawbacks with this system include its high cost and the time lag between the request for and the final delivery of the forms by a commercial vendor. Additionally, when changes are needed, the process for procuring forms must begin anew.

A system has been developed at USAFSAM in the Data Sciences Division which addresses these problems of cost and implementation time by enabling the creation of OPSCAN readable forms within the labs at USAFSAM. The time for design and implementation is significantly reduced. Also, the system is readily adaptable to change. The program is written in FORTRAN IV and implemented on an IBM 360/65. Appendix A contains a source listing of the program. Subroutines from the Calcomp Basic Software for the Calcomp 936 Plotter were used to generate the actual forms.

The system has been used in several USAFSAM projects including the Chamber Flight Records Repository, the Hyperbaric Medicine Division Data Base, the Toxic Fuel Compound Study, and the USAFSAM Acceleration Repository. Examples of these implementations are available from the authors.

GENERAL DESCRIPTION

The purpose of the program is to produce a form on the Calcomp 936 which can be used to print OPSCAN readable data collection forms. The layout of the form is designed within the restrictions of the optical scanner using the 'Symbol' and 'Line' subroutines from the plotter software package. The arguments for these subroutines are created by the program based on user-supplied instructions.

To understand how to supply instructions to the program, the user must first understand the format of the optical scan data sheet. Figure 1 (Chamber Flight Record) shows an example of an optical scan data sheet. The form illustrates the various types of data elements which can be coded. For simplicity, we shall refer to a data element as a field. The fields consist of either one box or groups of boxes which are labeled according to their function. OPSCAN boxes are the small rectangles found on all OPSCAN forms, where the data monitor makes his marks. For example, in the 'Reactor Data' sections of Figure 1 are three fields labeled "GL." There is only one box in each of these fields. Most of the fields, however, consist of rows of boxes, usually ten in length. Each box corresponds to 0,1,2,...9, one box for each possible response by a data monitor. It is not necessary, though, to have ten boxes in a row. Fewer than ten boxes are used in some instances, such as with the fields labeled "sex" and "race," each having two and three boxes respectively. Conversely, more than ten boxes can be grouped together if needed to make a field. For example, if a user needed a name block, each row would require 26 boxes, one for each letter in the alphabet. Often, several rows are grouped together to make one field which requires a larger response distribution. Two rows are grouped together for responses 00,..., 99. Three rows are grouped together for responses 000,...,999. Nine rows are grouped together for the Social Security Account Number field.

With the aid of an OPSCAN program sheet illustrated in Figure 2, fields can easily be placed on the sheets. The program sheet shows the placement for every possible OPSCAN box on the sheet. If the program sheet could be considered as lying in the first quadrant of a cartesian coordinate plane, each box could be identified with an ordered pair of integers. The numbers printed across the page in the columns can serve as the set of all possible abscissas. (Let P1 and P2 correspond to 0 and 1 respectively.) By numbering each row, from 0, starting with row 31A and proceeding up the page, the set of all possible ordinates is generated. If a field should consist of a single box, one ordered pair will uniquely identify the area. If a row of boxes is needed, two ordered pairs that identify the first and last box of the row are needed to identify the field. The program automatically draws those two boxes and all the ones in between them. If a block of boxes is needed, four ordered pairs which identify the corners will identify the whole field. Again, the program automatically draws all the boxes within the area outlined by the four ordered pairs.

The titling capabilities of the system are evident from Figure 1. All titling is done with the 'Symbol' subroutine available through the plotter software package. The titling over the fields is done with one call to the subroutine. The stacked titling which labels the boxes for easier marking is done by a series of calls to the 'Symbol' subroutine.

One additional capability of the system allows for drawing free form lines and rectangles. Drawing of lines is user controlled by

giving the program two ordered pairs, in inches, which are the coordinates of the endpoints of the desired line with respect to the lower left-hand corner of the program sheet. Rectangles can likewise be drawn by supplying to the program four ordered pairs which are the four corners of the desired figure. All straight lines and rectangles shown in Figure 1 were drawn using the 'Line' subroutine of the plotter software package.

Ancillary functions of the system include the ability to draw OPSCAN boxes with a break in the center to make room for lettering and the ability to draw all boxes, both with and without breaks in the center, slightly smaller. This option is useful when there are many fields that need to be defined on one page. By reducing the size of the OPSCAN boxes the finished sheet will not appear crowded as would the sheet with normal sized boxes.

All instructions, except for titling, are supplied to the program, through a single-card formatted record. Titling requires two records, one for the coordinates, and one for the actual title. Leading and imbedded blanks in the title are significant. Trailing blanks are ignored. All instructions read by the program are in the format: (8F9.4,T80,I4). The first eight fields give the coordinate measurements in inches for instructions 1, 2, and 3. For instructions 4, 5, 6, and 7 they refer to the position of the OPSCAN boxes on the program sheet relative to the lower left-hand corner as described earlier. Column 80 is reserved for a one-digit integer corresponding to the control instruction type. A summary of control instructions is given below:

(1) STANDARD TITLING

(2) SPECIAL TITLING

(3) STRAIGHT LINES AND RECTANGLES

(4) OPSCAN BOXES

(5) OPSCAN BOXES WITH BREAK IN THE CENTER FOR LETTERING

(6) SAME AS (4), ONLY SMALLER

(7) SAME AS (5), ONLY SMALLER

SAMPLE APPLICATION

To better understand how to generate an OPSCAN readable form, a sample application will be described. Let's say there is a user who is taking a survey of students' reaction to an educational film. The user has ten multiple-choice questions for the students to answer and would like to correlate their responses to their educational level and

age. Also, the user will need to have the students' names and social security account numbers (SSAN). Now, because of the type of application the user decides to collect data on OPSCAN data forms to be designed by the system described herein. He will collect the student name, SSAN, age, and educational level as well as the responses to the ten questions.

Where these fields are placed on the form is primarily a function of the user's needs as well as the inherent restrictions of the optical reader. Figure 3 shows a program sheet with the fields marked off by hand. Notice that the name field is 20 positions wide and 26 positions long. If any row is longer than 10 characters in length, each box in the field corresponds to the letter in the alphabet with the same relative column position as the box. If the field is 10 characters or less in length, as with the other fields in Figure 3, each box corresponds to the set of arabic numerals 0 - 9.

As was stated earlier, the fields can be identified by one or more ordered pairs. The fields for the ten responses and the educational level are single rows. Each can be uniquely identified with two ordered pairs. Using the numbering system for the program sheet described earlier, the fields can be identified with the following ordered pairs:

EDUCATIONAL LEVEL: (39,45) (42,45). Likewise, the ten response fields can be identified as follows:

RESPONSE 1: (4,37) (7,37)

RESPONSE 2: (4,34) (7,34)

RESPONSE 3: (4,31) (7,31)

RESPONSE 4: (4,28) (7,28)

RESPONSE 5: (4,25) (7,25)

RESPONSE 6: (12,37) (15,37)

RESPONSE 7: (12,34) (15,34)

RESPONSE 8: (12,31) (15,31)

RESPONSE 9: (12,28) (15,28)

RESPONSE 10: (12,25) (15,25)

The three blocks of rows for the social security number, the age, and the name fields can be identified by four ordered pairs corresponding to the four corners of each field, as follows:

NAME: (3,60) (28,60) (3,42) (28,42)

SSAN: (39,60) (48,60) (39,52) (48,52)

AGE: (39,49) (48,49) (39,48) (48,48).

By supplying all the aforementioned parameters to the system the graph shown in Figure 4 will be generated. Notice, that for the name, SSAN, age, and response fields, the boxes are drawn with a break in the center. This break accommodates the lettering seen in Figure 5. Notice that, with respect to the grid origin found in the lower left-hand corner of the figure, the letters are printed parallel to the x axis, yet each letter is rotated 90 degrees. This lettering is done with a series of calls to the 'Symbol' subroutine, one call for each letter. The user needs to supply only one instruction to the program, along with the actual title. The program automatically recalls the subroutine the appropriate number of times.

Additional titling as shown in Figure 6 is easily done. Each title requires one user-supplied instruction and the program calls the 'Symbol' subroutine only once for each title. The straight lines and rectangles seen in Figure 6 are also controlled by the program. The user supplies the endpoints of the lines and the corners of the rectangles to the system. Appropriate use of these capabilities helps the data transcribers in their work by making the form clear and easy to understand.

Figure 6 shows the final form for our sample application. Appendix B contains a complete listing of the input data.

CONCLUSION

A system has been developed at USAFSAM in the Data Sciences Division which creates a graph which can be used to produce OPSCAN readable data collection forms. Use of this system is especially beneficial when a potential user has a production application. The high speed processing afforded by optical readers along with free form design of the data sheets made possible by this system can enhance the design of the user's overall system.

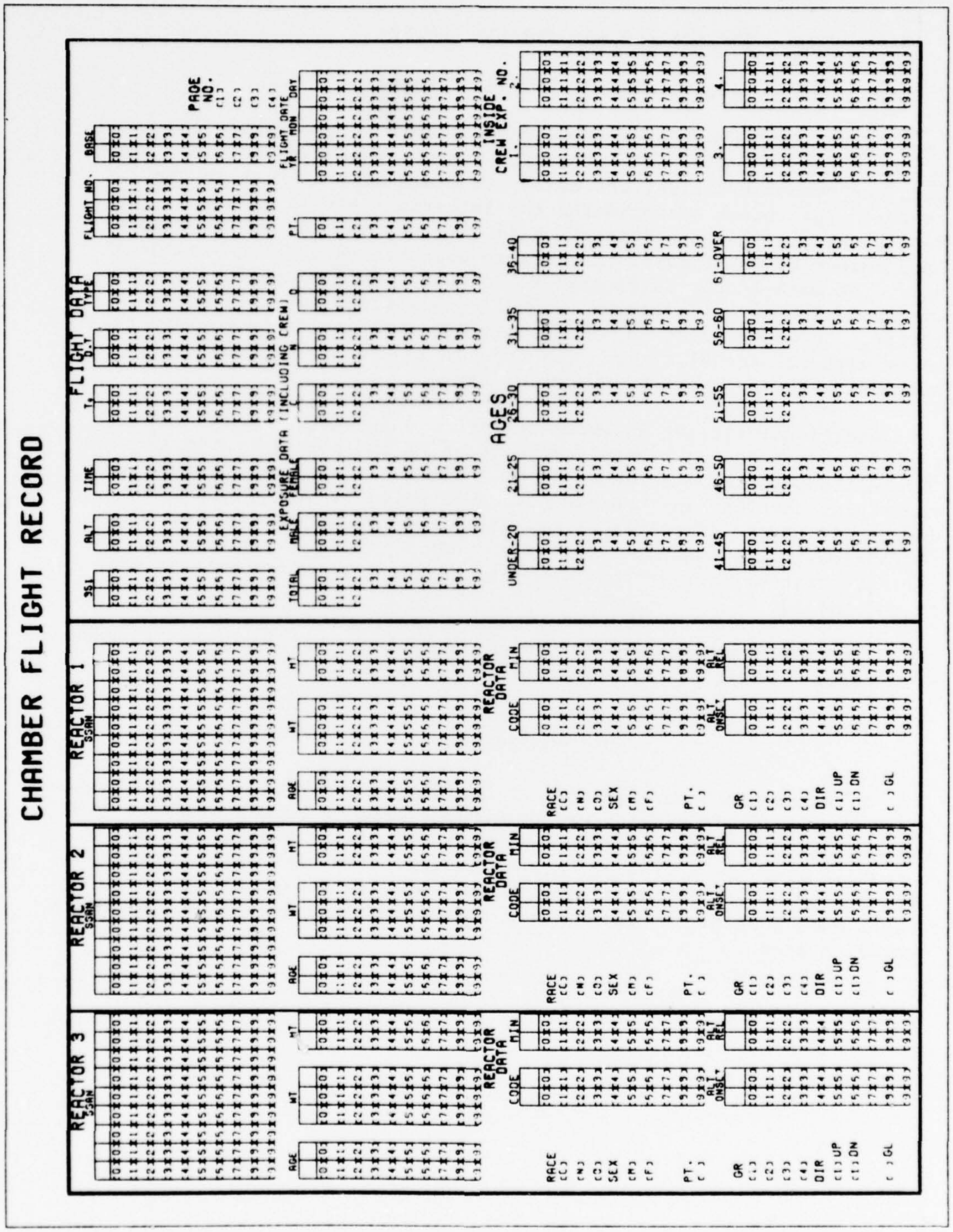


Figure 1. Sample OPSCAN data sheet.

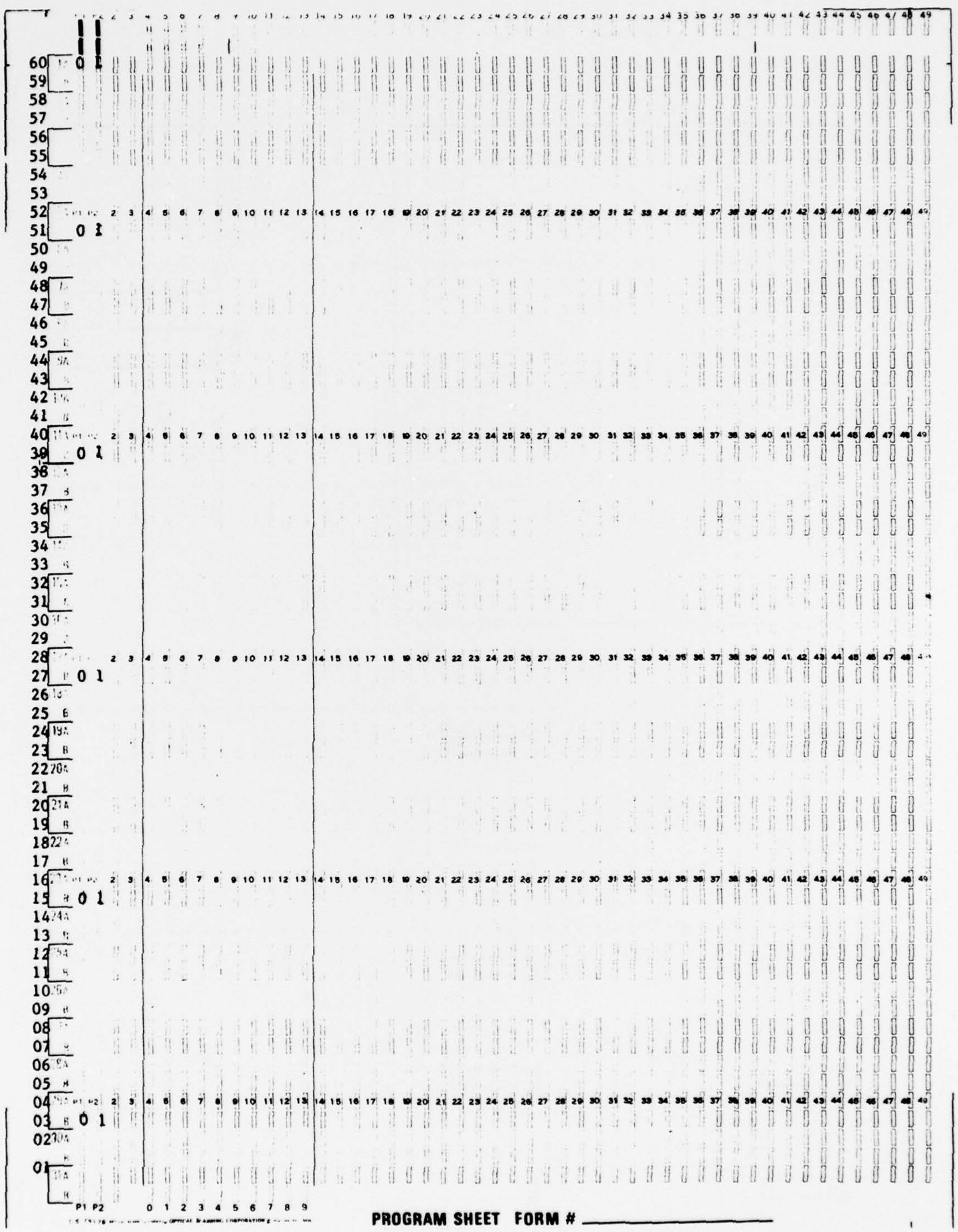


Figure 2. OPSCAN program sheet.

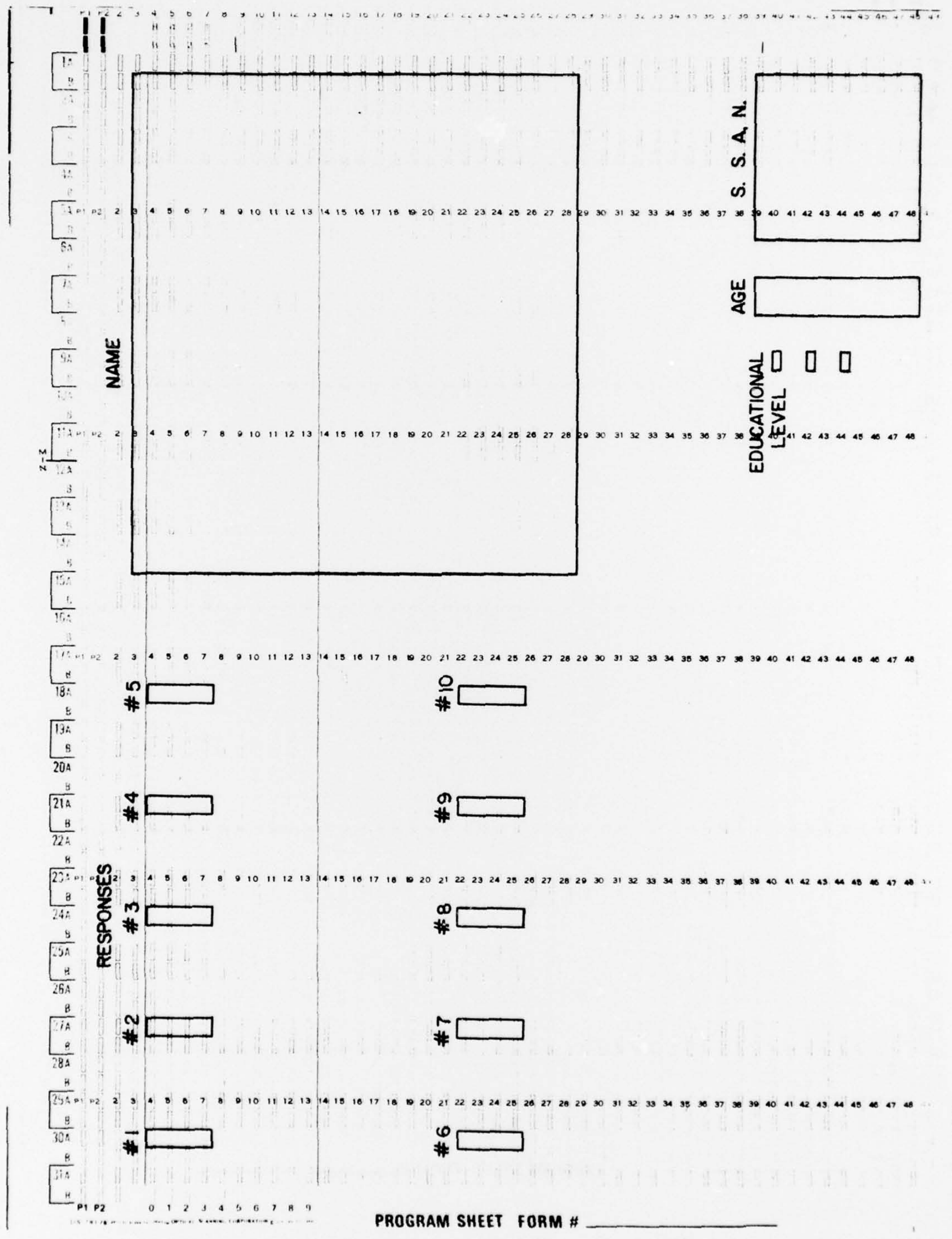


Figure 3. Field layout sheet.

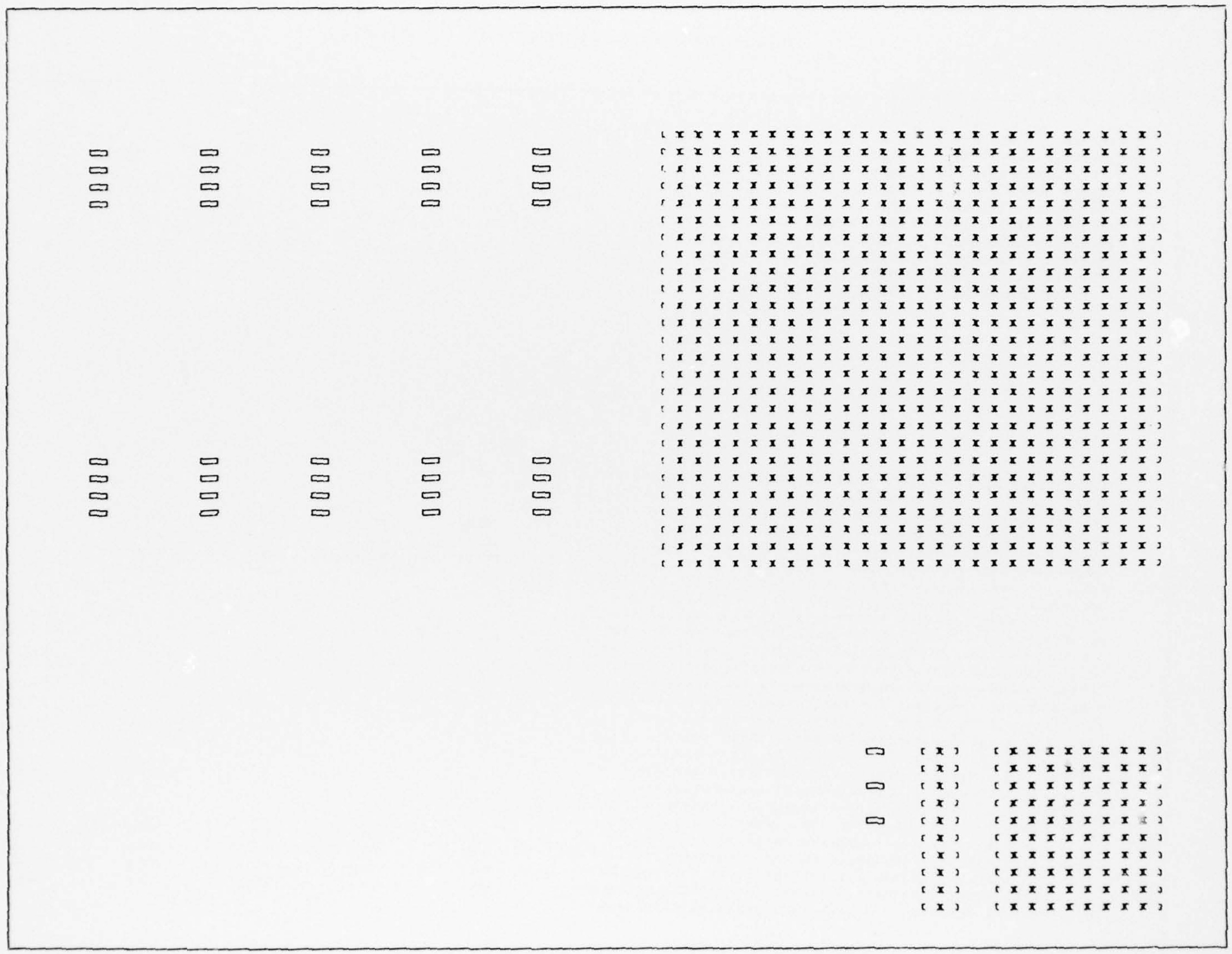


Figure 4. Initial output graph.

Figure 5. Intermediate output graph.

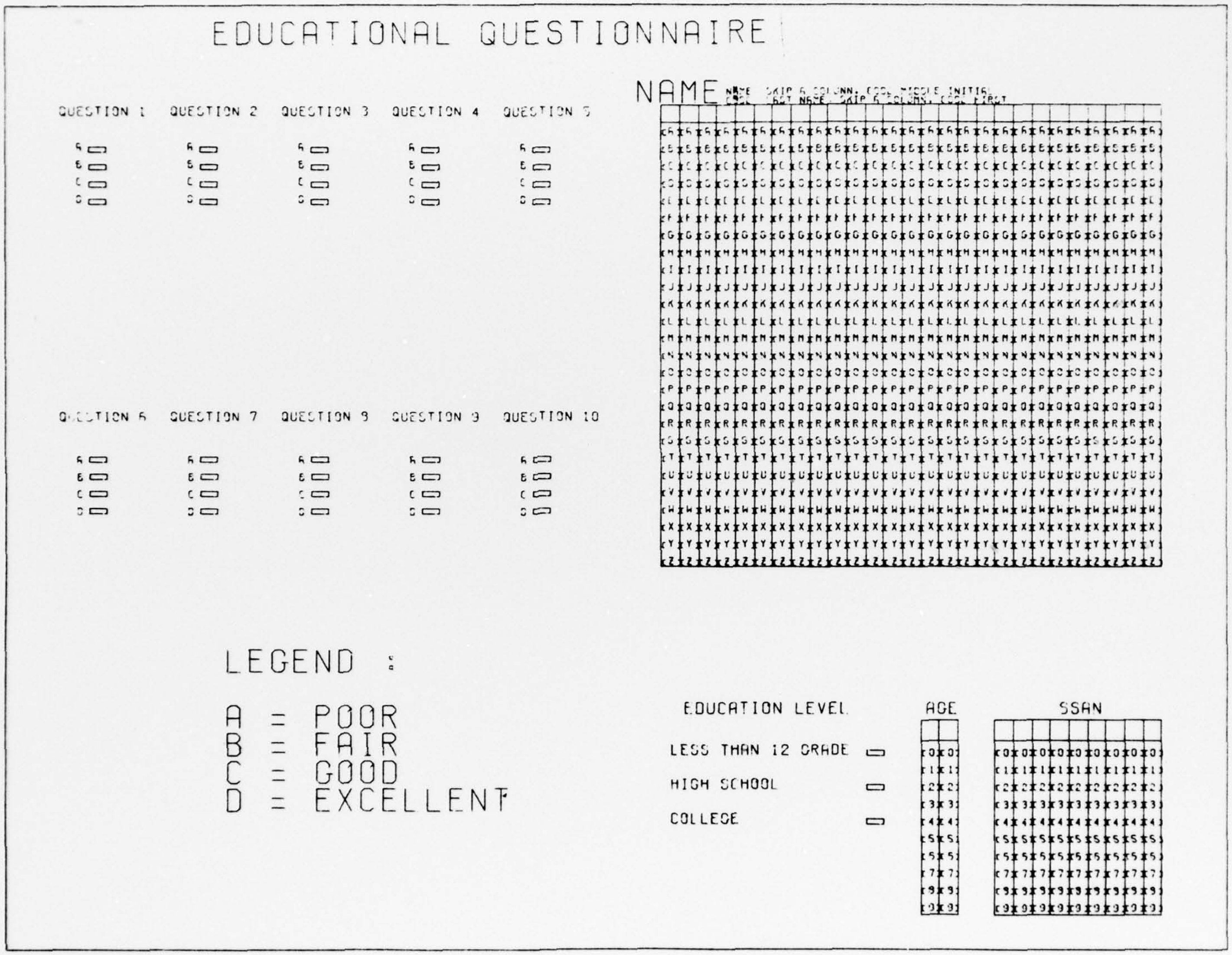


Figure 6. Final output graph.

APPENDIX A

SOURCE LISTING OF THE PROGRAM

```
C  THE FUNCTION OF THIS PROGRAM IS TO GENERATE INSTRUCTIONS THAT WILL
C  CREATE A PLOT, THROUGH THE USE OF THE CALCOMP PLOTTER. THIS OUTPUT
C  WILL BE USED TO GENERATE OPSCAN READABLE FORMS.
C
C  THE PROGRAM HAS SEVEN (7) TYPES OF INPUT AS FOLLOWS:
C
C   (1) STANDARD TITLING
C   (2) SPECIAL  TITLING
C   (3) STRAIGHT LINES AND RECTANGLES
C   (4) OPSCAN BOXES
C   (5) OPSCAN BOXES WITH BREAK IN THE CENTER FOR LETTERING
C   (6) SAME AS (4), ONLY SMALLER
C   (7) SAME AS (5), ONLY SMALLER
C
C       THE NUMBER IN PARENTHESIS IS THE INSTRUCTION CODE FOR EACH
C       FUNCTION
C
C
      INTEGER TITLE1,TITLE2
      DIMENSION XLCODE(7)
      DIMENSION XR(8),IN(8)
      DIMENSION XIN(8),X(7),Y(7),TITLE1(80),TITLE2(20)
      DATA XLCODE/0.,0.,0.,.153,.153,.0905,.0905/
C  INITIALIZE PLOTTER
      CALL PLOTS(0,0,8)
      CALL FACTOR(.8)
      CALL PLOT(0.,-14.75,-3)
      CALL PLOT(0., .00,-3)
C  READ IN 8 VALUES AND INSTRUCTION CODE
   15 READ (5,20,END=400) (XIN(I),I=1,8),ITYPE
   20 FORMAT (8F9.4,T80,I1)
C  DETERMINE INSTRUCTION CODE IN ORDER TO EVALUATE INPUT PARAMETERS
      IF (ITYPE - 2) 100,200,300
   10 READ (5,20,END=380) (XIN(I),I=1,8),ITYPE
      IF (ITYPE - 2) 100,200,300
C  *** STANDARD TITLING ***
C  BRANCH TO SUBROUTINE TO READ IN TITLE AND COUNT THE NUMBER OF
C   CHARACTERS
  100 CALL TIT  (TITLE1,TITLE2,NC)
C  USE STANDARD CALCOMP TITLING TO PLOT TITLE
      CALL SYMBOL  (XIN(1),XIN(2),XIN(3),TITLE2,XIN(4),NC)
C  BRANCH FOR NEXT INSTRUCTION
      GO TO 10
C  *** SPECIAL TITLING ***  -- USER NOT RESTRICTED TO STANDARD CALCOMP.
C                              PROGRAM PLOTS ONE LETTER AT A TIME.
C                              THIS MAY BE USED FOR STACKING TITLES.
C  BRANCH TO SUBROUTINE TO READ IN TITLE AND COUNT CHARACTERS
  200 CALL TIT   (TITLE1,TITLE2,NC)
      A = 90. +  XIN(4)
```

```
      XX= XIN(1)
      YY= XIN(2)
      XIN(5) = XIN(5) + .0003
C  PRINT ONE CHARACTER AT A TIME
      DO 290 I = 1,NC
      CALL SYMBOL (XX,YY,XIN(3),TITLE1(I),A,1)
      XX = XX + (XIN(5) + XIN(3)) * COS(XIN(4)/57.2957)
      YY = YY + (XIN(5) + XIN(3)) * SIN(XIN(4)/57.2957)
  290 CONTINUE
C  BRANCH FOR NEXT INSTRUCTION
      GO TO 10
C  *** LINES AND RECTANGLES ***
C
  300 IF (ITYPE - 4) 305,500,500
C
C       IS INSTRUCTION FOR LINE OR RECTANGLE?
  305 IF (XIN(5)  .EQ.  0.   .AND.  XIN(7) .EQ..0)  GO TO 350
C  * RECTANGLE *
C
C  POSITION PLOTTER
      CALL PLOT (XIN(1),XIN(2),3)
      I1  = 0
C  SETUP ARRAYS  FOR CALCOMP CALL
      DO 310 I = 1,8,2
      I1 = I1 + 1
      X(I1)   =   XIN(I)
  310 Y(I1)   =   XIN(I+1)
      X(5) = X(1)
      Y(5) = Y(1)
      X(6) = 0.0
      Y(6) = 0.0
      X(7) = 1.0
      Y(7) = 1.0
C  PLOT RECTANGLE
      CALL LINE (X,Y,5,1,0,1)
C  BRANCH FOR NEXT INSTRUCTION
      GO TO 10
C  * LINE *
C  POSITION PLOTTER
  350 CALL PLOT (XIN(1),XIN(2),3)
      I1 = 0
C  SETUP ARRAY   FOR CALCOMP CALL
      DO 375 I = 1,4,2
      I1 = I1 + 1
      X(I1)  = XIN(I)
  375 Y(I1)  = XIN(I + 1)
      X(3) = 0.0
      Y(3) = 0.0
      X(4) = 1.0
      Y(4) = 1.0
C  PLOT LINE
      CALL LINE (X,Y,2,1,0,1)
C  BRANCH FOR NEXT INSTRUCTION
      GO TO 10
C
```

```
C  *** CPSCAN SIZE BOXES ***
 500 DO 505 I = 1,8
C  ROUND OFF INPUT VALUES
     XR(I)  = XIN(I) +.5
 505 IN(I)  = XR(I)
C       IS INSTRUCTION FOR SINGLE BOX, ROW OF BOXES, OR BLOCK OF BOXES?
     IF (IN (3) .NE. 0  .OR. IN(5) .NE. 0) GO TO 600
C
C  * SINGLE BOX *
C
C  BRANCH TO SUBROUTINE TO MAKE ONE BOX
     CALL BOX (IN(1),IN(2),ITYPE,XLCODE(ITYPE))
C  BRANCH FOR NEXT INSTRUCTION
     GO TO 10
C       IS INSTRUCTION FOR ROW OF BOXES OR BLOCK OF BOXES?
 600 IF (IN(5) .NE. 0) GO TO 700
C
C  * ROW OF BOXES *
C
C  DETERMINE NUMBER OF BOXES AND BRANCH TO SUBROUTINE N TIMES FOR N BOXES
     N = IN(3) - IN(1) + 1
     DO 610 K = 1,N
     CALL BOX (IN(1) + (K-1),IN(2),ITYPE,XLCODE(ITYPE))
 610 CONTINUE
C  BRANCH FOR NEXT INSTRUCTION
     GO TO 10
C
C  * BLOCK OF BOXES *
C
C  PROGRAM HANDLES BLOCK OF BOXES AS GROUP OF CONTIGUOUS ROWS.
C  PROGRAM COUNTS NUMBER OF ROWS AND PLOTS EACH ONE JUST LIKE ABOVE.
 700 M = IN(2) - IN(6) + 1
     N = IN(3) - IN(1) + 1
     DO 710 K = 1,M
     DO 710 J = 1,N
     CALL BOX (IN(1) + (J-1),IN(6) + (K-1),ITYPE,XLCODE(ITYPE))
 710 CONTINUE
C  BRANCH FOR NEXT INSTRUCTION
     GO TO 10
 380 CALL PLOT (12.0,0.0, -3)
     GO TO 15
 400 CALL PLOT(12.0,0.0,999)
     STOP
     END
     SUBROUTINE BOX(IX,IY,NC,XLEN)
     DIMENSION X(7),Y(7)
C  SETUP CONSTANTS FOR PREDETERMINED BLOCK SIZE
     X(1) = IX  * .1551  + .6875
     Y(1) = IY  * .166667  + .5
     X(1) = X(1) - .0234375
     Y(1) = Y(1) - XLEN/2.
C       WILL BOXES HAVE BREAK IN CENTERS FOR TITLING?
     IF (NC .EQ. 4 .OR. NC .EQ. 6) GO TO 10
C  * BREAKS *  - BRANCH TO SEPARATE ROUTINE,USING SAME CONSTANTS
     CALL BRACK(X(1),Y(1),XLEN)
```

```
      RETURN
C
C  * NO BREAKS *
C  SETUP ARRAYS FOR CALCOMP CALL
   10 X(2) = X(1)
      Y(2) = Y(1) + XLEN
      X(3) = X(2) +. 046875
      Y(3) = Y(2)
      X(4) = X(3)
      Y(4) = Y(3) - XLEN
      X(5) = X(1)
      Y(5) = Y(1)
      X(6) = 0.0
      Y(6) = 0.0
      X(7) = 1.0
      Y(7) = 1.0
C  PLOT
      CALL LINE (X,Y,5,1,0,1)
      RETURN
      END
      SUBROUTINE BRACK(X1,Y1,XLEN)
      DIMENSION V(6),W(6)
C  SETUP ARRAYS AND PLOT HALF BOX
      V(1) = X1
      W(1) = Y1 + XLEN/5
      V(2) = X1
      W(2) = Y1
      V(3) = X1 + .046875
      W(3) = W(2)
      V(4) = V(3)
      W(4) = W(3) + XLEN/5
      V(5) = 0.0
      W(5) = 0.0
      V(6) = 1.0
      W(6) = 1.0
      CALL LINE (V,W,4,1,0,1)
C  SETUP ARRAYS AND PLOT OTHER HALF
      W(1) = W(1)  +(XLEN - 2*(XLEN/5))
      V(2) = V(1)
      W(2) = W(1)  + XLEN/5
      V(3) = V(2) + .046875
      W(3) = W(2)
      V(4) = V(3)
      W(4) = W(3)  - XLEN/5
      V(5) = 0.0
      W(5) = 0.0
      V(6) = 1.0
      W(6) = 1.0
      CALL LINE (V,W,4,1,0,1)
      RETURN
      END
      SUBROUTINE TIT   (TITLE1,TITLE2,NC)
      INTEGER TITLE1,TITLE2,BLNK
      DIMENSION TITLE1(80),TITLE2(20)
```

```
      DATA  BLNK/'    '/
C  READ TITLE TWICE -SEPARATE FORMATS NEEDED FOR COUNTING CHARACTERS
C                     AND STORING FOR CALCOMP
      READ(5,105) TITLE1,TITLE2
  105 FORMAT (80A1,T1,20A4)
      NC = 80
      DO 120 I = 1,80
      I1 =  80 -I + 1
      IF (TITLE1(I1) .NE. BLNK) GO TO 125
      NC = NC - 1
  120 CONTINUE
  125 RETURN
      END
```

APPENDIX B

DATA LISTING FOR SAMPLE APPLICATION

```
   3.       59.       28.       59.        3.       33.                            5
  39.       59.       48.       59.       39.       51.                            5
  39.       48.       48.       48.       39.       47.                            5
  39.       44.                                                                    4
  41.       44.                                                                    4
  43.       44.                                                                    4
   4.        2.        7.        2.                                                4
   4.        8.        7.        8.                                                4
   4.       14.        7.       14.                                                4
   4.       20.        7.       20.                                                4
   4.       26.        7.       26.                                                4
  22.        2.       25.        2.                                                4
  22.        8.       25.        8.                                                4
  22.       14.       25.       14.                                                4
  22.       20.       25.       20.                                                4
  22.       26.       25.       26.                                                4
   1.17     5.98      .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     6.144     .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     6.312     .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     6.48      .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     6.6469    .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     6.81      .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     6.98      .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     7.1540    .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     7.3125    .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     7.4831    .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     7.6469    .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     7.8187    .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     7.99      .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     8.15      .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     8.3209    .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     8.49      .0625     0.0        .0922                                   2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     8.64      .0625     0.0        .0922                                   2
```

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     8.8150    .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     8.99      .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     9.1579    .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     9.32      .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     9.48      .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     9.6401    .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     9.8075    .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17     9.98      .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17    10.14      .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   1.17    10.31      .0625   0.0       .0922                                2
ABCDEFGHIJKLMNOPQRSTUVWXYZ
   6.75     8.99      .0625   0.0       .0922                                2
0123456789
   6.75     9.1579    .0625   0.0       .0922                                2
0123456789
   6.75     9.32      .0625   0.0       .0922                                2
0123456789
   6.75     9.48      .0625   0.0       .0922                                2
0123456789
   6.75     9.6401    .0625   0.0       .0922                                2
0123456789
   6.75     9.8075    .0625   0.0       .0922                                2
0123456789
   6.75     9.98      .0625   0.0       .0922                                2
0123456789
   6.75    10.14      .0625   0.0       .0922                                2
0123456789
   6.75    10.31      .0625   0.0       .0922                                2
0123456789
   6.75     8.49      .0625   0.0       .0922                                2
0123456789
   6.75     8.3209    .0625   0.0       .0922                                2
0123456789
   1.3125    .65      .0625   0.0       .0922                                2
ABCD
   1.3125   1.65      .0625   0.0       .0922                                2
ABCD
   1.3125   2.65      .0625   0.0       .0922                                2
ABCD
   1.3125   3.65      .0625   0.0       .0922                                2
ABCD
   1.3125   4.65      .0625   0.0       .0922                                2
ABCD
   4.125     .65      .0625   0.0       .0922                                2
ABCD
```

4.125	1.65	.0625	0.0	.0922	2
ABCD					
4.125	2.65	.0625	0.0	.0922	2
ABCD					
4.125	3.65	.0625	0.0	.0922	2
ABCD					
4.125	4.65	.0625	0.0	.0922	2
ABCD					
.906	5.9062	.906	10.4062		3
1.0625	5.9062	1.062	10.4062		3
.906	5.9062	5.062	5.9062		3
.906	6.0781	5.062	6.0781		3
.906	6.25	5.062	6.25		3
.906	6.4062	5.062	6.4062		3
.906	6.5781	5.062	6.5781		3
.906	6.75	5.062	6.75		3
.906	6.9062	5.062	6.9062		3
.906	7.0781	5.062	7.0781		3
.906	7.25	5.062	7.25		3
.906	7.4062	5.062	7.4062		3
.906	7.5781	5.062	7.5781		3
.906	7.75	5.062	7.75		3
.906	7.9062	5.062	7.9062		3
.906	8.0781	5.062	8.0781		3
.906	8.25	5.062	8.25		3
.906	8.4062	5.062	8.4062		3
.906	8.5781	5.062	8.5781		3
.906	8.75	5.062	8.75		3
.906	8.9062	5.062	8.9062		3
.906	9.0781	5.062	9.0781		3
.906	9.25	5.062	9.25		3
.906	9.4062	5.062	9.4062		3
.906	9.5781	5.062	9.5781		3
.906	9.75	5.062	9.75		3
.906	9.9062	5.062	9.9062		3
.906	10.0781	5.062	10.0781		3
.906	10.25	5.062	10.25		3
.906	10.4062	5.062	10.4062		3
5.0625	5.9062	5.062	10.4062		3
6.4375	8.25	8.1875	8.25		3
6.4375	8.4062	8.1875	8.40622		3
6.4375	8.5781	8.1875	8.57812		3
6.4375	8.9062	8.1875	8.90622		3
6.4375	9.0781	8.1875	9.07812		3
6.4375	9.25	8.1875	9.25		3
6.4375	9.4062	8.1875	9.40622		3
6.4375	9.5781	8.1875	9.57812		3
6.4375	9.75	8.1875	9.75		3
6.4375	9.9062	8.1875	9.90622		3
6.4375	10.0781	8.1875	10.07812		3
6.4375	10.25	8.1875	10.25		3
6.4375	10.4062	8.1875	10.40622		3
6.4375	8.25	6.4375	8.57812		3
6.625	8.25	6.625	8.57812		3
6.4375	8.9062	6.4375	10.40622		3

```
   6.625     8.9062    6.625     10.40622                                        3
   8.1875    8.25      8.1875     8.5781                                         3
   8.1875    8.9062    8.1875    10.4062                                         3
   3.75      0.5        .08      90.        0.                                   1
QUESTION 6
   3.75      1.5        .08      90.        0.                                   1
QUESTION 7
   3.75      2.5        .08      90.        0.                                   1
QUESTION 8
   3.75      3.5        .08      90.        0.                                   1
QUESTION 9
   3.75      4.5        .08      90.        0.                                   1
QUESTION 10
   1.0       0.5        .08      90.        0.                                   1
QUESTION 1
   1.0       1.5        .08      90.        0.                                   1
QUESTION 2
   1.0       2.5        .08      90.        0.                                   1
QUESTION 3
   1.0       3.5        .08      90.        0.                                   1
QUESTION 4
   1.0       4.5        .08      90.        0.                                   1
QUESTION 5
   6.75      6.0        .09      90.        0.0                                  1
LESS THAN 12 GRADE
   7.06      6.0        .09      90.        0.0                                  1
HIGH SCHOOL
   7.37      6.0        .09      90.        0.0                                  1
COLLEGE
   0.35      1.69       .20      90.0       0.0                                  1
EDUCATIONAL QUESTIONNAIRE
   6.0       2.0        .20      90.        0.0                                  1
LEGEND :
   6.50      2.0        .20      90.        0.0                                  1
A = POOR
   6.75      2.0        .20      90.        0.0                                  1
B = FAIR
   7.00      2.0        .20      90.        0.0                                  1
C = GOOD
   7.25      2.0        .20      90.        0.0                                  1
D = EXCELLENT
   0.8125    6.5        .06      90.        0.0                                  1
NAME, SKIP A COLUMN; CODE MIDDLE INITIAL
   0.8925    6.5        .06      90.        0.0                                  1
CODE LAST NAME, SKIP A COLUMN; CODE FIRST
   0.8825    5.7        .20      90.        0.0                                  1
N[illegible]
   0.8825    5.7        .20      90.        0.0                                  1
NAME:
   6.38      9.5        .10      90.        0.0                                  1
SSAN
   6.38      8.30       .10      90.        0.0                                  1
AGE
   6.38      6.125      .10      90.        0.0                                  1
EDUCATION LEVEL
```

USAF SCHOOL OF AEROSPACE MEDICINE
AEROSPACE MEDICAL DIVISION (AFSC)
BROOKS AFB TX 78235